# Introduction

The increasing demand on facility management, inspection, and maintenance operations has caused an urgent need to integrate building information models and geographic information systems models. Consider the following real world example: someone made a request for maintenance to look at the temperature in “core chem Physics office A249”, since it was too cold (14 degrees). The operators need to use a campus building operation system to locate the particular building and room cited in the work request. Then they check the heating and cooling systems that serve that particular area or room. In addition to the location of the equipment (such as the heating system) in the building, the maintenance personnel may need information about the manufacturer, serial number, maintenance history information, service manual, and spare part information about the specific equipment that needs to be maintained or repaired or replaced. In order to create such a comprehensive campus operation system to retrieve both building information and explore the relationships among buildings in a campus area, we need the system to contain both the geospatial campus data and a maximum level of detail about the real buildings. Currently, we do not have any single model or system that incorporates both of them; they exist only in separate models.

There are many different building information models and 3D geospatial information models representing real world surrounding objects from both geometrical and semantic perspectives. Today, Industry Foundation Classes (IFC) and City Geography Markup Language (CityGML) are two of the most prominent semantic models; IFC is used for design, and CityGML is used for describing built building infrastructures [10]. IFC [5] was introduced as a standard for describing building components and construction data. CityGML [7] is a common information model and XML-based encoding system for the representation, storage, and exchange of virtual 3D city and landscape models. The integration of IFC and CityGML is seen as a necessary step for getting a complete 3D city model with detailed building information. The reason for this is that this integration would assist us in handling the operation and maintenance requests by exploring the 3D campus view to locate the desired buildings (using CityGML) and zoom into the building to display the layout of every floor so that we can easy identify the room that stores the equipment that needs to be checked and/or repaired (using IFC).

Generally speaking, all the previous integration approaches to the implementation of a semantic and geometrical integration of IFC and CityGML models [12][14][15] have focused mainly on unidirectional conversion from IFC to CityGML, because CityGML is more popular for decision-making as it has a broader view. They define some standard mapping schemas between IFC and CityGML based on the semantic agreement of their interpretation of the same objects. There are conversion tools [15] which extend CityGML with rich semantic information of IFC and ADE. In addition, commercial software products for conversion from IFC to CityGML, such as IfcExplorer [18] and Safe software [19], make a great effort to the development of 3D city modeling integration.

Our goal in this thesis was to build an operation and maintenance system to allow users explore and retrieve a wide range of campus data, starting from campus scale and zooming into a room in a building. To allow this integration, we need to be able to simultaneously query IFC and CityGML. CityGML represents geospatial as well as the building components. On the other hand, IFC only has the detailed building information, excluding the outer surrounding environment. After considering the previous conversion approaches [12][14][15][16] and commercial software products [18][19][21], we choseto build our integration system on top of the existing work. We selected two most suitable frameworks as a start point of our conversion from IFC to CityGML, BIMServer [21] and Feature Manipulation Engine (FME) [23].

After comparing and analyzing the converted results from BIMServer and FME, we decided to extend the CityGML schema to accommodate the rich semantic building information that the existing conversion tools cannot cover. After running several experiments, we concluded that it is not currently possible to bring all the details for buildings such as the mechanical components and facility information into CityGML. We encountered a number of problems in our attempt: (1) the standard CityGML schema cannot currently support such kinds of information , and (2) file size increases dramatically when small amounts of additional information is added in. The file size is far beyond the maximum capability that the current conversion applications can handle.

After we realized that the conversion from one model to another model, no matter what direct it is, is infeasible to implement, we switched our approach to incorporate them together into one central system. There are two common systems to combine different data sources into one central destination, data integration and data warehouse. After thoughtfully considering their difference and respective usages, we decided to implement a data integration system to corporate IFC and CityGML schemas regard to their overlapping information, which is able to retrieve information from both of them on a simple and uniformed interface. We will explain its architecture and detailed implementation procedure in the late sections.

In the remainder of this thesis, section 2 explains the terminologies and concepts used in our project, such as the IFC, CityGML, the schema in different standard formats, for instance Extensible Markup Language (XML) and relational model. We first introduce several applications we explore in our project for the management and visualization of 3D city model and the transformation between different models in section 3. Section 4 introduces the previous and ongoing research work and commercial products of conversion frameworks and tools. In section 5, we introduce data integration and data warehouse respectively, make a comparison of their strengths and weaknesses in terms of storage size and construction time and explain the best circumstances they apply individually. Section 6 and section 7 demonstrate the process we extract the IFC schema and CityGML schema in relational format from their standard schema specifications respectively. Next, we explain the process of establish our data integration system by steps and the challenges remaining in our project in section 8. After that, we present our experiment result in section 9, explore other ongoing research topics in the geographic information field in section 10 and make a conclusion in the last section.

# Background

As we discussed above, to fully handle maintenance and operation requests, we need to access a complete picture of 3D city modeling at high levels of detail. In this section we review the Building Information Model (BIM) [2] in detail in Section 2.1 and then discuss the higher-level CityGML [4] in Section 2.2.

## BIM and IFC

“Building Information Modeling (BIM) is a digital representation of physical and functional characteristics of buildings.”[3] A BIM describes buildings with respect to their geometric and semantic properties. Generally, it is generated at the early stage of the building’s lifecycle to facilitate the architects, civil engineers and stakeholders in planning, designing and constructing. It has the ability to organize huge volumes of data related to buildings, the semantic information of building parts and spatial relationships between them, and also supports sophisticated 3D visualization and manipulation. Unlike CAD models, which represent buildings as a collection of points and lines, the semantic information that BIMs carry makes great contributions to the data analysis and decision making regard to buildings. It is an object oriented building modeling, which defines building components as elements and their properties and the relationships between them.

One of the most developed and established semantic models that implement BIM concepts is the Industry Foundation Classes (IFC) [5]. As an open standard schema, IFC is popularly used to exchange and share BIM data between different applications. Its standard schema comprises information contributing to a building’s whole lifecycle: from conception, through design, construction and operation to maintenance and destruction [6]. IFC is a conceptual model for buildings which represents building structural components and their relationships semantically as shown in the below Unified Modeling Language diagram in Figure 1. It describes the components of spatial objects as classes and different arrows means different relationships and association between classes. IFC has an IfcBuilding class which consists of one or more IFCBuildingStoreys. In each IFCBuildingStorey, there are several IfcSpaces instead of rooms. The building elements are walls, roofs, beams, columns, stairs, also contains openings, such as doors, windows.

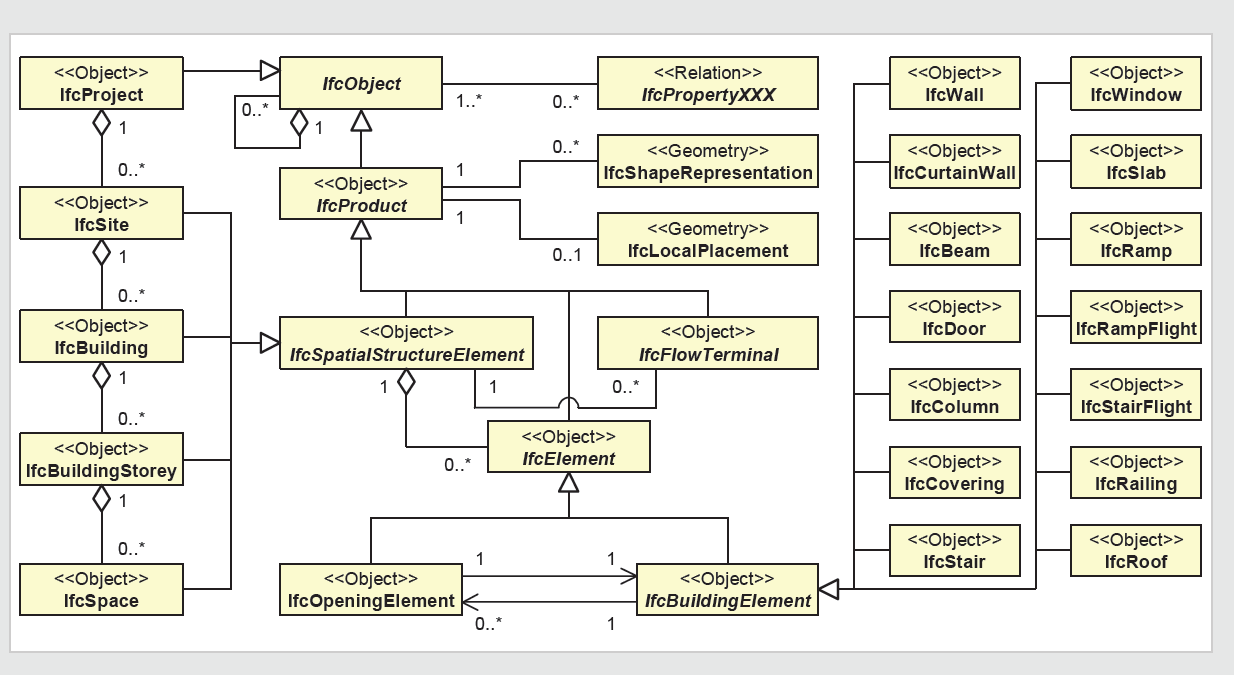
**

Figure 1: A UML notation of IFC building model.

## CityGML

CityGML [7] is defined as a semantic information model, and it is used as a standard representation to store and exchange virtual 3D objects and city models among different applications. As a Geospatial (GIS) system, CityGML uses five different levels of detail (LoD), which are used to represent the same city model objects in different degrees of detail, where objects become more detailed with the increasing LoD regarding both geometry and thematic differentiation:

* LoD0 represents a two and half dimensional model of urban surfaces in which buildings are represented as footprints. As shown in the Figure 2, it is a topographic model of bare-earth terrain which displays the elevations of natural terrain features, such as barren ridge tops and river valleys [28]. However, elevations of the buildings and roads are digitally removed, so buildings are represented as 2D rectangle shapes as you can see from the top down.
* LoD1 is the first representation of buildings as 3D objects. All buildings look like blocks with the same flat roofs.
* In LoD2 and LoD3, a building is represented in an architectural model with details of roofs, walls, some exterior elements, such as balconies, and openings in the boundary surfaces, such as doors or windows. Lod3 applies more details and facade textures to the roofs and walls than LoD2, and it covers detailed vegetation and transportation objects which cannot be found in LoD2. [9]
* The most detailed level, LoD4 [9] adds detailed interior elements to buildings, such as rooms, interior doors, interior wall surfaces, stairs, furniture, electricity units.

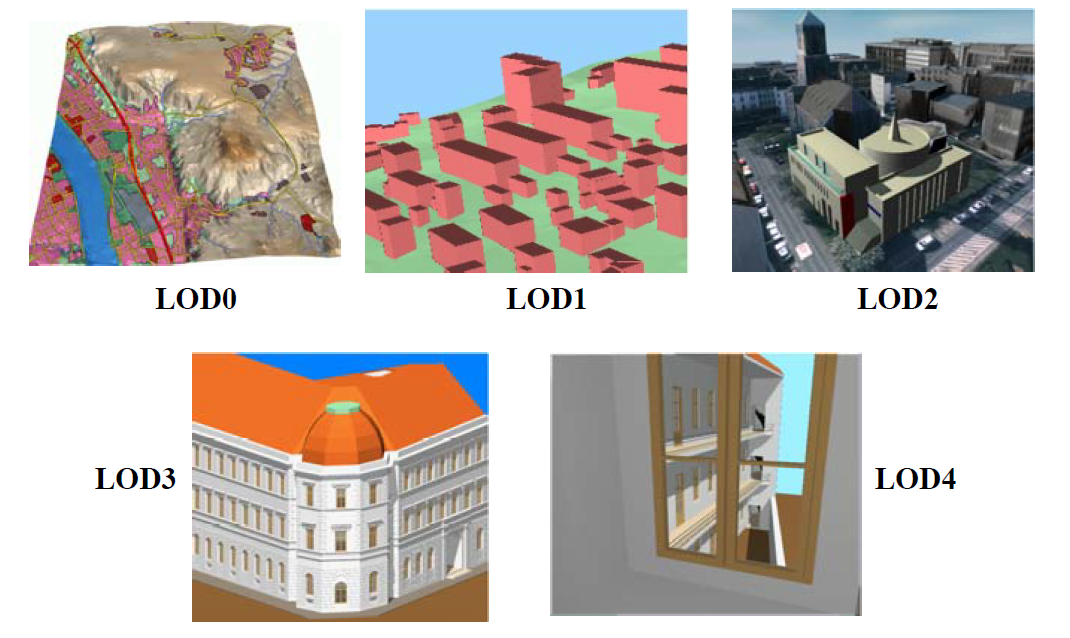


Figure 2: The five levels of detail (LOD) of CityGML (Source: IKG Uni Bonn [9]).

Thanks to these different levels of detail, CityGML is highly scalable, so the building details can be flexible in order to be adjusted to meet projects’ special needs and in order to make efficient and sophisticated analysis. Compared to IFC, in Figure 3 CityGML does not have storey information, and it considers rooms as building components rather than spaces in IFC. CityGML represents the geographic information of spatial objects; thus, the measurement properties of each building component are represented as geographic coordinates instead of lengths and widths.

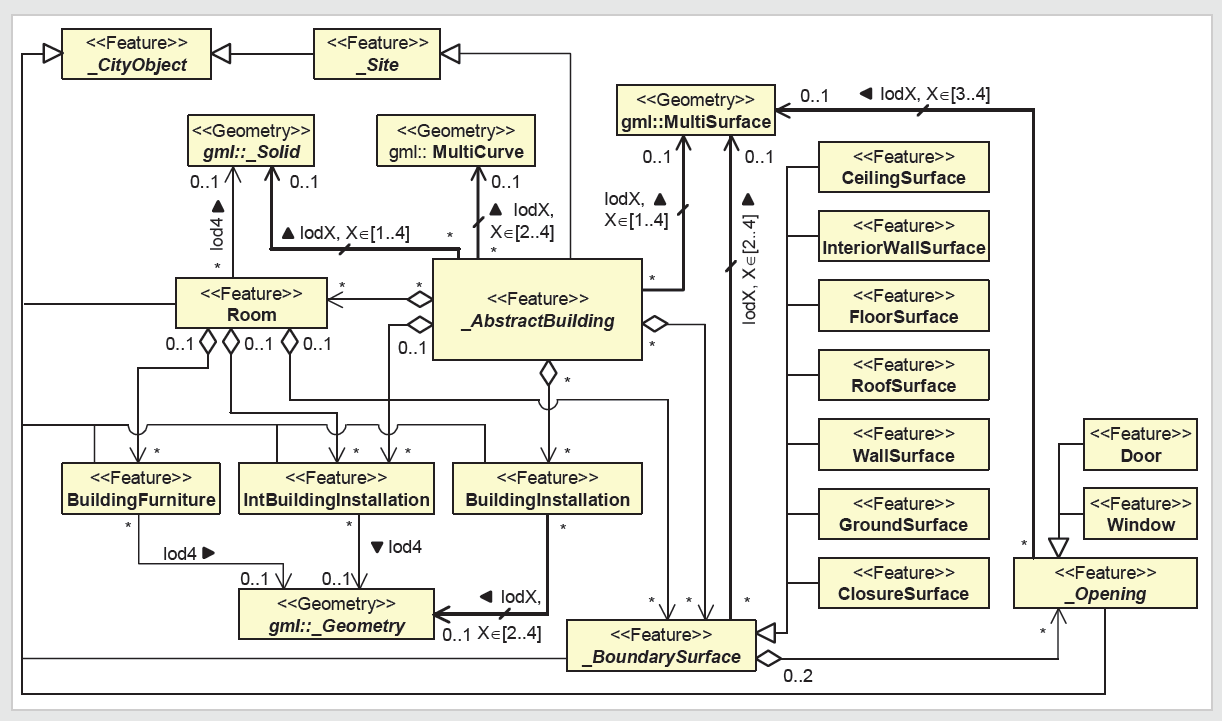


Figure 3: A UML notation of CityGML Model.

Generally, [8] IFC, an element-based volume model, uses constructive solid geometry with volumetric and parametric primitives representing the structural components of buildings. However, a 3D GIS, a surface model, uses boundary representation, which is the accumulation of observable surfaces of topographical features.

Right now, FME Data Inspector supports the visualization of the IFC and CityGML files. The Solibri Model Checker provides sophisticated functions to analyze the BIM in the IFC format. Also, Galdos GML Inspector and LandXplore can view the CityGML data.

## Basic database concepts and definitions

“A schema is a collection of entities (or classes), attributes, and relationships between entities.” [30] A schema defines the logical structure of the database; its formal definition language depends on the type of the database system. A schema consists of a set of . Each relation represents an entity or a relation between entities with a set of attributes to describe it. A relation is expressed in a format of R(Attr1, . . . , Attr n), in which R is the name of the relation, Attrx is an attribute at the position x and the relation has number of n attributes.

“A model is the population of a schema, following the patterns, templates and constraints stipulated by the schema. It contains the actual instances of the entities (or classes)” [30].

Therefore, IFC and CityGML’s standard specifications are schemas, but the IFC and CityGML files describing the real spatial objects are models.

Datalog is a [declarative](http://en.wikipedia.org/wiki/Declarative_programming) [logic programming](http://en.wikipedia.org/wiki/Logic_programming) language, which has a format of

head :- subgoal1, subgoal2, …, subgoal…. It has two parts separated by “:-”, the left part is the head, and the right part is the body that consists of a group of subgoals. The head and subgoals are all relations, each of which has a name and a set of variables associated with it. The subgoals in the body can be placed in any order. Datalog is always used to declare queries, because the variables in the head can be expressed from the body, and it supports the basic query operations of select, project, and join. This is also called a conjunctive query. If the same variable name appears at multiple places in the body, it is a joining variable used to link several tables together.

## XML

Unlike relational databases with fairly well-structured forms as tables, Extensible Markup Language (XML) is “a very popular standard format which [is being] used widely right now and…is a simple very flexible text format [that] is playing an increasingly important role in the exchange of a wide variety of data on the web and elsewhere” [54]. XML Schema Definition (XSD) is a standard language used to define XML schema, which describes the XML document structure and content in a brief and simple way. XML has a flexible structure, called semi-structured, because, while some of its content can come from part of the schema, it can express other information not in the schema. Compared to the relational schema, in which a relation is a set of attributes used to describe a concept or a relation in parallel, XML allows users to store heterogeneous data, in which relations can be hierarchically structured, while also allowing multiple values for one element. Therefore, its semi-structured format contributes to its expressiveness and flexibility, which not only **facilitates data exchanges among different formats, but also allows for the storage of rich and diverse information.**



Figure 4: Part of the XML text document from [rap' slice].

Figure 4 displays part of an XML text document of two people’s names, addresses and phone numbers. The element names are enclosed within tags, such as “<person>”, and the values are set between element tags, like “Mary”. The heterogeneous structure allows for a hierarchy to be nested inside one element, such as, for example, the “street”, “no”, “city” elements are nested within the element “address”, which belongs to the “person” element. Since it is not necessary that every person has a phone number, the address format can vary for different people.

On the other hand, because of XML’s loose structure; many traditional database techniques cannot be applied to it to improve its performance, such as an index, for example. Since it lacks the classic query optimization, its syntax-parse based query execution algorithm is much less efficient than a relational database. Unlike the relational database, in which the data is stored in tables, XML is not intended for human beings to read and its redundant tag representation costs overwhelming storage waste. Additionally, it is difficult to detect improper format and/or syntax errors.

# Popular applications used in civil engineering

As there are many geospatial data manipulation applications in the market today, in this sectionwe explore some of them, compare their functions and choose some appropriate applications to apply in our project to help us make better decisions about operation and maintenance requests. We look at these applications’ analyzing, navigation, and visualization functions, and describe four representative applications in detail below. While they all play an important role in the 3D geographic information field, they all have different focuses. For instance, some of them can only support one model, either BIM or GIS. Some are designed for transformation between different models, and others are very powerful navigation tools of the 3D city.

## LandXplorer

The LandXplorer product developed by Autodesk is “a powerful software system for the management and visualization of geovirtual 3D city models and 3D landscape models” [44]. Not only does LandXplorer provide various functionalities that can explore, analyze, query, and navigate a 3D virtual city, but it also presents a fundamental raster-based digital terrain model, on top of which it has additional geospatial data (buildings, plants, transportations) in various data formats like GIS and CAD, which are imported and integrated into the city model, as ‘city models’ and 3D objects. LandXplorer can separately represent every LoD of the CityGML model by importing and applying each layer of texture and appearance gradually onto the basic LoD1 city model. LandXplorer offers an exploring panel, which provides a hierarchical view of the spatial objects and their attributes in the city model. In addition, it also supports searching and query functions to display desired buildings or spatial objects that meet certain criteria. For example, we can ask LandXplorer to display all the buildings built before a certain year or in a certain area. Furthermore, LandXplorer supports a connection to the 3D city database to import, export, and merge CityGML data.

Because of LandXplorer’s sophisticated navigation ability and its query function, our initial intension was to translate all possible data models into CityGML and then import them into LandXplorer to execute complex queries and analyses.

## Revit Architecture

Also developed by Autodesk, the Revit Architecture product [47] is specifically designed for Building Information Modeling (BIM), and is employed throughout the whole lifecycle of a building, from conception, design, construction and operation, to maintenance and destruction. Because Revit is made only for buildings, it does not take the building’s surroundings into account. One of the most popular software programs, Revit is widely used in architectural design, as it facilities the architects’ jobs by providing accurate and high-quality user-interfaces and tools to build BIM workflows of design, documentation, and construction. Revit also benefits civil engineers by providing architectural design solutions and collaboration platforms among building project teams. In addition, it supports evaluation functions to assess their designs and constructability. Finally, it integrates separate parts of the BIM model together to complete the building’s entire lifecycle, which assists architects and designers in making better decisions.

Revit is designed to server different types of customers, such as architects, designers, engineers, construction managers, and stakeholders, so it provides several different kinds of views to them, respectively, such as an entire 3D model or one floor view or just one component. Finally, Revit is designed to eventually export to BIM in different data types, such as CAD, IFC, and some building models in XML (gbxml). In our project, we want the BIM to be exported in IFC or ifcXML formats.

## Feature Manipulation Engine

The Feature Manipulation Engine (FME) is “an application providing unrivalled format support for data translation and integration, and unlimited flexibility in data model transformation and distribution” [45]. It can read, analyze, translate, manipulate, and write spatial data in various geospatial data formats.

FME Desktop consists of the Universal Viewer, Data Inspector, Workbench, and Quick Translate components. While the Universal Viewer and Data Inspector can provide visualization of various data models, it cannot manipulate them. Universal Viewer can only handle 2D data and is the precursor to Data Inspector. Data Inspector, which can handle 2D and 3D geospatial data, was invented to replace Universal Viewer in the future, but it does not currently have all of the capabilities of Universal Viewer.

Quick Translate offers fast and simple translation between different data formats to facilitate the interoperability of spatial data, and the translation rules are all fixed without any customization. Unfortunately, its translation function is not sophisticated enough to achieve our goal.

Workbench is one of the most important and frustrating applications we have been using in our project. A typical workflow consists of deciding what data to read in (readers), how to translate the data (transformers), and what format to write out to (writers). The later section discusses the various transformations we have attempted to use to translate IFC data into the CityGML model, along with anevaluation of the results.

## **3D City Database Import/Export**

3D City Database, which is based on a standard spatial relational database system (Oracle Spatial Database) is used to store, represent, and manage virtual 3D city models [46]. It can accommodate “semantically rich, hierarchically structured, multi-scale urban objects facilitating complex GIS modeling and analysis tasks, far beyond visualization” [46]; furthermore, it supports five different levels of detail of CityGML representation and complex geometric and appearance data.

3D City Database is designed to connect between spatial data and 3D City database by the high-performance importing and exporting of a virtual 3D city model in the format of CityGML by following the CityGML specification/schema. It also supports the CityGML appearance model and XML validation of CityGML files. The Import/Export tool allows users to specify the objects to be imported or exported by filters and parameters. Its matching tool merges data from different data sources and different levels of details by eliminating the duplicate spatial objects and transferring geometrical information [46]. 3D City Database plays an important role in our project as a database to store the CityGML model, and its schema is utilized in the data integration algorithms.

## Solibri Model Checker

Solibri Model Checker is a software application developed by Solibri, Inc., “the leader in Model Checking technology and innovation for design quality assurance and BIM analysis” [35]. Recently, it released **Solibri Model Checker v8.1, which improves user experience and enhances the “**availability and accessibility of BIM data and visual information” [35]. As you can see below, it provides an easy-to-use visualization interface; the left panel displays all the elements in a building represented in a tree hierarchy structure and the right part demonstrates a complete 3D view of a building.

In our project, the IFC schema is generated from Solibri’s attribute sets with respect to each spatial object. The reason we use this is that it represents information exactly following IFC’s standard specifications and provides all the information we want for our project. There are some other similar tools, such as IFC Engine Viewer [36], a free IFC browser developed by TNO, Germany. However, Solibri Model Checker covers more information and provides more powerful visualization and analyzing functions.

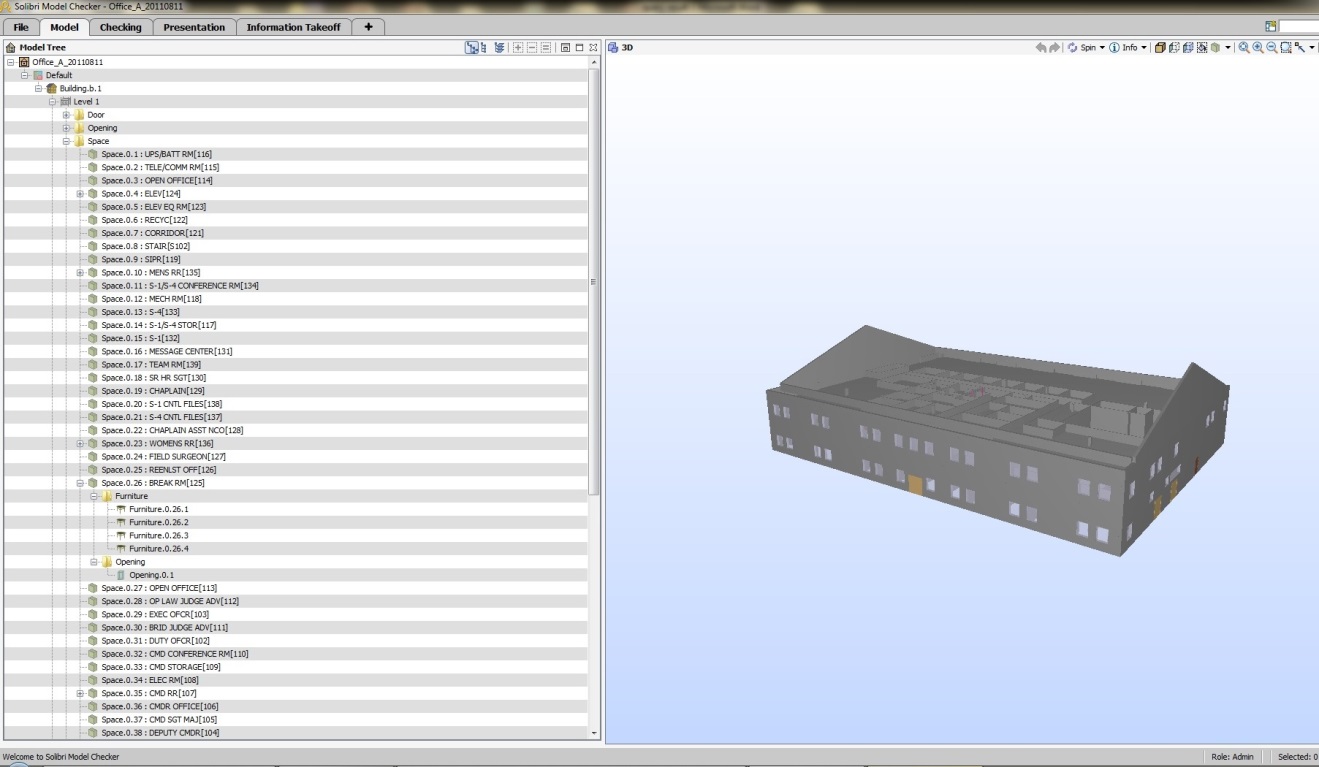


Figure 8: Solibri interface

# Existing Conversion Approaches

In order to process the maintenance requests, such as the previous example’s uncomfortable room temperature in the “core chem Physics office A249”, we need to show not only the map of the entire campus, which indicates the paths to locate that building, but also the details of each floor in the specified building, so the maintenance worker can access the room where the heating system is located. While building the GIS model of the whole campus is considered a good starting point, we can then increase the picture by including the detailed building information from the BIMs. In sum, for all the operation requests, the most important thing we should do is to define the scale and region that we will operate at. We believe the best solution would be the integration of the BIM and the GIS to present the behavior of the entire campus and also the details of each individual building.

“Researchers all agree that the best approach for the integration of BIM and GIS is harmonized semantics” [22], which create formal mappings between the detailed inner building models and outdoor real world GIS. Both the IFC and CityGML use different terminologies to describe the same domain. Semantic interoperability ensures that both IFC and CityGML share the same meaning for a defined spatial object. All the previous approaches are based on the idea of harmonized semantics; some of them focus on a unidirectional method (mostly from IFC to CityGML) for the conversion process. While part of this attempt is to develop an automatic framework to transform the IFC building models into CityGML with regard to each level of detail, another is to convert building information to GIS from CAD instead of BIM. Bidirectional conversion has recently drawn our attention as a means of fully integrating IFC and CityGML in terms of a conceptual model.

Some of the most significant existing approaches to the integration of IFC and CityGML are listed below:

**The IFG Project**

The framework of the IFC for the GIS (IFG) project (in 2003) [11] was designed to exchange building information between CAD systems and GIS, using IFC, so that it could support the analysis of the relationships among a building’s different areas or other buildings, by accessing to the details of both the buildings and their surrounding environments. The project succeeded in creating mapping specifications from the IFC geometry to the GIS, and vice versa, by identifying entities in the IFC schema that coulld support the GIS application and creating mappings between them and the real GIS entities.

**Nagel’s framework of conversion of IFC to CityGML**

In 2007, Nagel proposed a framework [12] that aimed at developing algorithms that automatically transform IFC building models into CityGML models, through a series of steps that create the separate footprints of each storey within their own boundary surfaces and finally merge them together. As this research focused only on level of detail (LOD)1 and LOD2 of CityGML, the purpose of the algorithms was to create a geometrically and semantically valid representation of LOD1, which could also be applied to LOD2.

**Isikdag and Zlatanova’s advanced framework**

Isikdag and Zlatanova (2009b) [14] have extended Nagel’s framework by proposing a framework for automatic generation of building semantics and components in CityGML from their BIM representation. Because the CityGML and IFC models are designed for two different domains, they have very diverse objects and classes and cannot be directly and easily mapped to each other. Therefore, they generate semantic and geometric mappings for each LoD of CityGML separately, since the same object in one schema can be mapped to different objects in the other for different levels of detail. In order to simplify and facilitate the conversion process, for each of level of detail, all the objects and attributes in both schemas that need to be transformed are first defined, as are the mapping rules.

**A 3D Conversion Framework by Thomas Kolbe**

With the aim of creating a holistic view of a 3D city, a team led by Thomas Kolbe at the Technical University of Berlin [13] proposed a framework that incorporates semantic spatial context data into 3D graphics/data of buildings and urban areas stored in formats such as X3D, DXF, KML, and COLLADA. The reason we chose CityGML as the intermediate layer for the conversion process to IFC is because it is a very rich semantic model that shares a similar notion of building semantics with the IFC. Therefore, we considered it a good bridge to link 3D visualised models to the sophisticated and detailed building models. The simple summary of this conversion framework is shown below:

Geometric/graphics Models 🡪 Semantic City Model 🡪 Building Information Model.

Their corresponding formats are:

X3D, DXF, KML, COLLADA 🡪 CityGML 🡪 IFC

**The development of the GeoBIM -- CityGML extension**

In 2009, Léon [15] demonstrated the latest application domain extension (ADE), which can convert the building information model (BIM) in the format of the open standard IFC into a CityGML (van Berlo, 2009). Because CityGML represents building information in a high level of detail, in order to represent the rich semantic information of IFC in CityGML, the researchers extend CityGML schemas with extra objects and properties. However, there are only few IFC classes that can be transformed in to CityGML extensions and have real meanings in it, because IFC and CityGML describe buildings in different representations and aspects.

**Unified Building Model**

The Unified Building Model (UBM) [16][17] represents the first fully integrated framework of IFC and CityGML in which IFC can be traced to CityGML and vice versa. The reference ontology in this study is defined as an expressive ontology for IFC and CityGML semantic models, which is a superset model that is extended to contain all the features and objects from both the IFC and CityGML building models, with respect to all levels of detail, including inner and outer spatial structures. The integration approach is performed in two steps: a building model is first converted from the source model into a UBM, which is then converted, from the UBM into the target building model. The UBM is considered as a standard schema, which generates mappings from both data sources to the standard schema. This standard schema can easily be extended if there is a demand for transformation from a new schema to an IFC or CityGML.

**Commercial software products**

Safe Software Company [19] developed several applications that support the management, exchange, and visualization of spatial and non-spatial data in more than 255 different formats. The feature manipulation engine (FME) is their core data transformation product, which supports the translation, transformation, and integration of spatial data. Right now, it supports the direct conversion of IFC to CityGML and vice versa.

Another commercial software product that converts from IFC to CityGML is IfcExplorer [18][20], which is an implementation of Nagel’s conversion algorithm by the Research Centre Karlsruhe, at the Institute for Applied Computer Science in Germany. IfcExplorer is designed to make an automatic conversion of an IFC model into a CityGML, with regard to the selection of the specified LODs and building elements.

**Building Information Model server**

Besides being free and open source, the intention of the Building Information Model server (BIMserver) software [21] is to centralize information from any building related project. It uses IFC as its core standard building model and stores building information in the format of IFC in an underlying database, so it is possible to query, merge, and filter all the BIM models and generate IFC files on the fly. And because it also supports exporting functionality in various formats, including CityGML, we consider it as a conversion tool for converting from IFC to CityGML. The increasing research work that contributes to the integration of IFC and CityGML is driven by the demand for exchanging and sharing spatial information among different applications, some of which are focused on buildings, while others address a wide range of city areas. To summarize, the contributions of all the existing research on the integration of IFC and CityGML are: they all define semantic mappings between IFC and CityGML; [14][15][16] develop frameworks and algorithms to make an automatic bidirectional or unidirectional conversion, in terms of geometric and semantic information; the integration is done with regard to each level of details of CityGML[12][14]. [15] also discusses the rich semantics of the IFC and how they can be applied to more detailed CityGML models.

The BIM and GIS are intended to serve two different domains and scales: the former focuses on inner building structural components, while the latter is used for mapping the surrounding real outer world, and, since they use different representations to describe the same spatial objects, they have very diverse objects, classes, and properties. In order toachieve an accurate and efficient integration, most of the previous works only focus on the main structural components, such as walls, roofs, doors, windows, and so on, of which IFC and CityGML have the same semantic and geometric representations. Even though some entities can be semantically mapped from IFC to CityGML, it is still difficult to create the geometric matching relations between them. For example, for a beam that runs across two rooms in CityGMLis represented as two thematic objects because it is observable from both rooms, but in IFC, they are aggregated into one object. Because of the different geometric representations that IFC and CityGML use, sometimes researchers [13] apply an evaluation function on all the possible ambiguous conversions in order to come up with an optimal one. Since the CityGML is capable of representing detailed building information in, at most, LoD4, and is extended to model noise, tunnels, bridges, hydro, and utility networks [27], it still cannot represent the mechanical elements. In addition, during the conversion process, the properties and parameters attached to the components need to be kept in the target schema, but CityGML is not capable of keeping them. Under the above restrictions, the complex schemas and components are beyond the scope of the current research, as well as any components with complicated geometric shapes.